

Modeling 3-D Complex Buildings With User Assistance

S. C. Lee, A. Huertas and R. Nevatia

*Institute for Robotics and Intelligent Systems
University of Southern California
Los Angeles, California 90089
{sungchul/huertas/nevatia}@iris.usc.edu*

Abstract

An effective 3D method incorporating user assistance for modeling complex buildings is proposed. This method utilizes the connectivity and similar structure information among unit blocks in a multi-component building structure, to enable the user to incrementally construct models of many types of buildings. The system attempts to minimize the time and the number of user interactions needed to assist an existing automatic system in this task. Several examples are presented that demonstrate significant improvement and efficiency, compared with other approaches and with purely manual systems.

1. Introduction

An important task of computer vision systems is to generate 3D models from images. In this paper we deal with 3D modeling of complex buildings from images of aerial scenes where it is not feasible or easy to control the environmental and viewing conditions. While a great deal of progress has been made in the development of methods and techniques for automated model construction the performance of complete systems remains below that of humans. Many systems however could exhibit significant increase in performance and effectiveness with minimal but critical human assistance. In this paper we present recent work that illustrates this approach for the task of modeling complex building structures from aerial images; a difficult but important task with many commercial and military applications.

Many challenges remain in the automatic construction of models of buildings from aerial scenes because of the broad scenarios and dimensions of complexity associated

with the domain. Building structures are present in rural, semi-urban and urban environments with increased levels of concentration and with increased complexity in shape. The connectivity among the structures in the forms of roadways and sidewalks introduce large numbers of “distractors” that contribute ambiguities that must be resolved. Also, the ability to extract significant geometric features depends on the quality, resolution and number of images available for the analyses. The viewpoint, illumination conditions and the time of the year also may affect significantly the appearance of the structures in terms of contrast and occlusions.

The results that can be achieved automatically are not completely accurate although significant progress has been made in recent years [1, 2, 3, 4]. On the other hand completely manual systems require an unacceptable amount of effort of time and cost. The increased availability of imagery provided at higher resolutions, and by several sensor modalities render these laborious tasks practically unfeasible. In this paper we describe new methods, using an approach that we have been developing over the last few years [5, 6], to include modeling of buildings having complex shapes. This approach attempts to minimize the user assists to an automatic system that is capable of modeling simpler shapes robustly, while maintaining a suitable collection of geometric features and relationships among them. The system also maintains full knowledge of the viewing parameters and illumination conditions, and allows the user to model much more complex structures with few interactions, interacting with a single view of the scene.

2. Approach

Several approaches to user-assisted modeling have been proposed. The conventional approach is to provide a set of generic models that are then fit to the image data by changing model and viewing parameters [7]. In this approach, the system provides geometric computations but substantial time and effort are required from the user.

* This research was supported in part by the Defense Advanced Research Projects Agency of the U.S. Government under contract DACA 76-97-K-0001, and in part, by the U.S. Army Research Office under grant No. DAAH04-96-1-0444.

Newer approaches have attempted to combine user input with varying amounts of automatic processing. In [6], the authors suggest providing just an approximate building location to extract a building but the quality of the final result is completely dependent on the automatic analysis. In [8], other interactive tools are described including methods for replicating model buildings that are identical or very similar to others. In [9], an automatic system constructs topological relations among 3D roof points collected by a user for each roof; this system can work with several types of complex roofs. In [10], the system handles complex building structures by using constructive solid geometry. This system uses an image correlation method to fit a primitive to the image; however, this method is computationally expensive when modeling urban sites, where many building have complex shapes.

In our approach, basic modeling tasks are performed by the underlying automatic system but this system receives critical assists from the user. The underlying system is the multi-view building detecting system described in [4]. This system assumes that buildings have rectilinear shapes, rectangular roofs, simple gabled roofs, and vertical walls. The capabilities of our user-assisted system however are not restricted to these shapes; it can assist the user in handling multi-component structures and non-rectilinear shapes.

The automated system forms rectangular hypotheses, in 3D, for the rectilinear portions of the buildings using lines, junctions, parallels, and U-structures that have been matched across two or more views of the scene. A verification step analyzes the evidence of shadows and visible vertical walls to determine valid hypotheses. Many of these hypotheses lack sufficient support from image evidence but a single interaction from the user could verify these hypotheses. When available, these rectilinear portions become “seed” models that, with user assists, evolve into models of complex shapes.

The user interaction consist of pointing and clicking (a mouse or similar device) on a feature (junction or line), on a display of one image of the scene. While other systems usually require interacting with more than one view, and the use of 3D cursors or pointing devices, our system uses only one working view. The underlying automatic processing however uses at least two images. Any available view can be used as the working view as all the views and models are geo-registered in 3D. The necessary projections are performed automatically depending on the view chosen by the user.

The capabilities of the underlying automatic system are restricted to rectilinear shapes and earlier versions of the user assisted system made use of many of its capabilities for hypothesis formation and validation [5]. The assisted system described here however, deals with buildings that

have arbitrary shapes, and therefore relies only on the lines and junctions extracted and matched among several views to generate 3D models directly from user inputs.

3. Methodology

Many multi-component buildings consist of a combination of rectangular-shaped roof sections as illustrated in Figure 1. These components are typically extracted separately by the automatic system. As a result, earlier versions of our assisted system [5] could assist the user in modeling this type of structures with results similar to that shown in Figure 2a. This approach required up to three clicks, plus a calculation of height, per component. For the example shown, that was 9 clicks in 7.5 seconds including three height calculations. The methodology in the current system allows incremental modeling with the result shown in Figure 2b; it required 5 clicks in 5.5 seconds including a single height calculation.

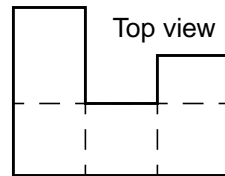
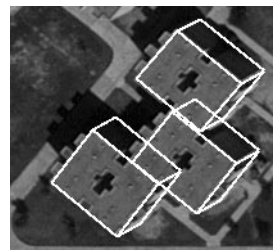
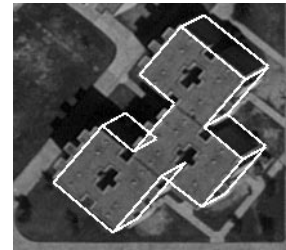


Figure 1. Complex building



(a) No connectivity



(b) Using connectivity

Figure 2. A Building with connected components

A more complete example of the same area is shown in Figure 3. Seven buildings, labeled “A”, were detected automatically. Nine “seed” components (3 clicks each building) and twenty two unit “blocks” (2 clicks each) were added by the system with user assistance to model the buildings labeled “B”. This task required 71 clicks and 85.2 seconds of wall time.

To help evaluate this result, the number of required clicks could be compared with the total number of corner points on the user-assisted modeled roofs; 150 corner points in this example, thus a 52% reduction in effort. If we also include ground points of buildings (for manual height adjustment), even more improvement can be achieved. A comparison, in terms of number of clicks and wall time, among the current method, an earlier versions of the system [5], and a manual method [7] is given in

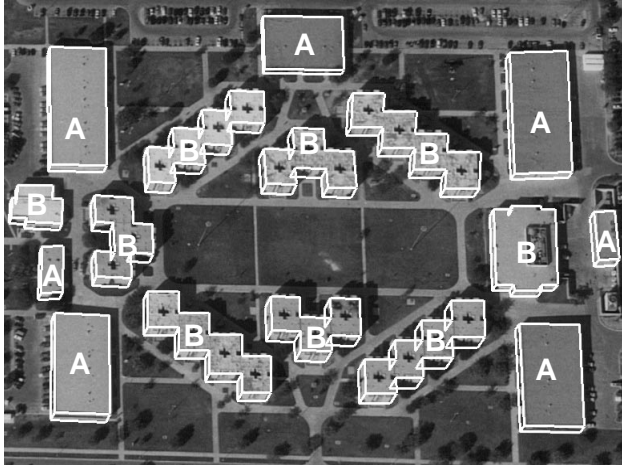


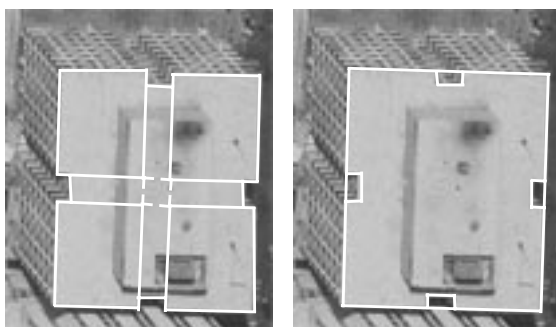
Figure 3. Results from one area of Ft. Hood Site: 71 clicks and 85.2 seconds

Table 1. The numbers shown do not include the clicks nor the time needed to select items from menus, which tend to be more heavily used in traditional manual systems.

TABLE 1. Comparison with other system

| Method | Clicks | Time (sec.) |
|-----------------|--------|-------------|
| New Method | 71 | 85.2 |
| Previous Method | 93 | 111.6 |
| Manual Method | 166 | 676.0 |

A second aspect of the general methodology to reduce interaction is the ability to subtract or remove components. The example shown in Figure 4a requires six added components (18 clicks) with the earlier system while the result in Figure 4b requires one addition and four subtractions (11 clicks.).



(a) Without subtracting (b) With subtracting
Figure 4. Combining add/subtract operations

4. Details of the method

User interactions consist of pointing and clicking near object junctions and boundaries on a single image. The system however uses multiple views to compute automatically 3D information based on line and junction features

extracted from at least two views. These features are matched among the available images and a record is made of the possible correspondences between them and their computed heights. Figure 5 shows the underlying matched lines and junctions corresponding to one of the views of our example of Figure 4.

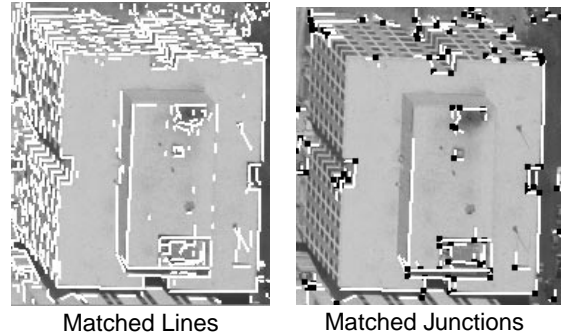


Figure 5. Underlying features

The user interactions (clicks) relate the click locations on the image to the underlying features. All the information associated with this features is then used to initiate or update the 3D models under construction.

The method consists of processes to initiate the construction of a model, and to add or subtract 3D blocks to it with minimum interaction and maximal use of automatically precomputed 3D information. A model starts with a “seed”. Seeds are generated in two ways: by running the automatic modeling system up to its capability, or by the user giving one click or by three clicks[5]. A seed is generated by one click if there is sufficient underlying evidence to form a 3D block. Otherwise, a second, and up to a third click are needed. Each one of these first two clicks (near a corner) causes the system to attempt to create a seed based on the underlying features. A single junction and its branches is sufficient to generate a seed block. The third click, if needed, always generates a seed. Multiple hypotheses are possible at each step and the system selects among them automatically by analyzing the matching evidence in all the available views. Note that the matching information provides the necessary 3D information.

Next, the user adds or subtracts blocks by clicking on roof locations. The height of the incremental models is derived automatically. All calculations are in 3D, with the appropriate transforms applied automatically for immediate viewing and storage of the results in a CAD-like form.

4.1. Multi-component buildings

A multi-component building, in our system is restricted to have all of its components have sides parallel to those of the seed component. With this property, only two clicks are needed to add or subtract a rectangular block. Additions and subtractions can be carried out in any order and with-

out specific ordering of the sequence of interactions. The system determines automatically whether a block is being added or subtracted from the current configuration of the model and from the user interactions with respect to the model.

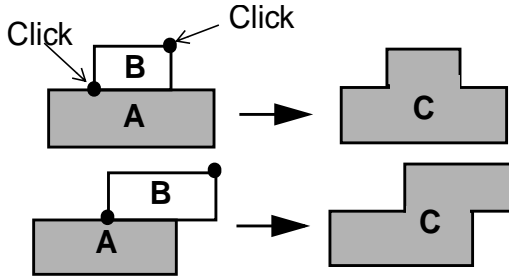


Figure 6. Protrusions are added by two clicks

Figure 6 illustrates how the user interaction adds a rectangular block **B** to block **A** by two clicks. One of the clicks is given on the existing outline on block **A** and the other diagonally across block **B**. As a result a new polygon **C** is computed. Figure 7 illustrates a similar process for indentations. These operations can be repeatedly applied until the resulting polygon correctly models the building roof.

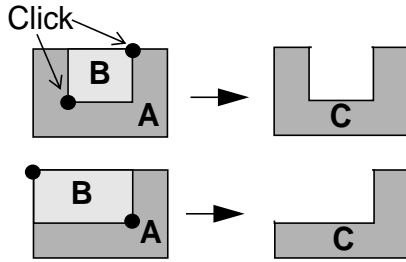


Figure 7. Indentations are added by two clicks

The following seven elements are involved in the update of the roof outline (also see Figure 8).

L_1 : A roof boundary line passing through P_1

L_2 : A roof boundary line intersecting L_1

L_3 : A parallel line with L_2 and passing through P_1

L_4 : A parallel line with L_1 and passing through P_2

L_5 : A parallel line with L_2 and passing through P_2

P_3 : Intersection point between L_3 and L_4

P_4 : Intersection point between L_1 and L_5

To determine the location of P_3 and P_4 , we first find the line L_1 among the boundary lines of the nearest roof. Next, we determine the necessary elements, lines and intersections needed to compute the location of (P_3 and P_4). The insertion of the computed points (P_3 and P_4) points along the boundary must preserve the polygonality of the bound-

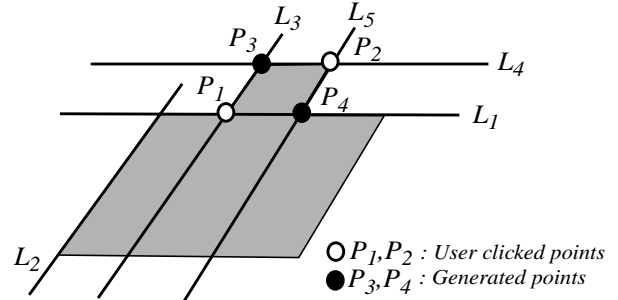


Figure 8. Adding a parallelogram to a polygon given two points, P_1 and P_2

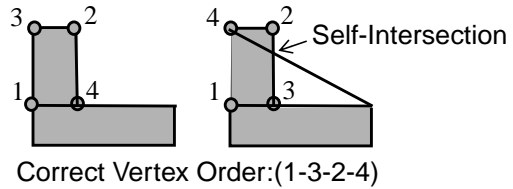


Figure 9. Self intersection

ary, i.e. the appropriate ordering of the vertices along the new boundary (the boundaries of roofs are stored as ordered sets of vertices). In some cases there is ambiguity in the selection of L_1 , for instance when P_1 coincides with a boundary junction, more than one alternative are possible. These are evaluated to select the configuration that do not include self intersections (intersections among a polygon boundary lines). Figure 9 gives an example of a self intersection. We discuss self intersections further below in Section 4.4.

Note that the system is generating an increasingly complex boundary in a precise and consistent manner. It enforces the parallel relationships, liberating the user from the need of strenuous accurate clicking, thus helping improve performance. Figure 10 shows an example of a building having indentations and a structure on the top. The seed component is shown in Figure 10a. After eight clicks of user interaction, the four unit blocks are subtracted, as shown in Figure 10b.

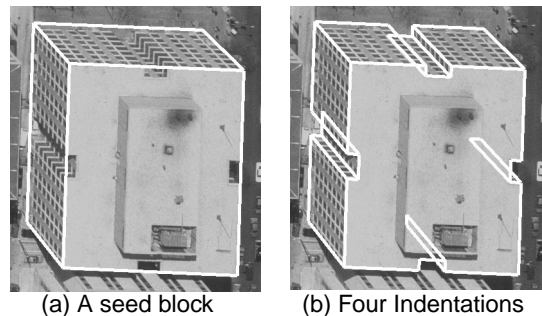


Figure 10. Multi-component building result

4.2. Non-rectangular buildings

The method described in the previous section applies to buildings whose corner elements are at, or close to, 90° in 3D. In order to construct models having non-rectangular roofs, the system uses triangular block operations in addition to the rectangular block operations discussed earlier. To add or subtract a triangular block we consider two cases:

In the first case only one click is needed to add or subtract a triangular block as illustrated in Figure 11a and Figure 11b. To add a triangular block the user clicks on, or near, a corner point. The linear edges that found near the location of the user's click, intersect the boundary of the seed roof. Two points out of all intersection points and the user clicked point form a triangular block to be added to the roof outline. Since multiple hypotheses are possible a verification process (see Section 4.4) is applied to select one. To subtract a triangular block (Figure 11b) the user clicks on, or near, a roof edge. The corresponding underlying line is extended until it intersects the boundary of the current roof at two points. The two intersection points and the user click point determine the triangular block to be removed from the model.

The second case arises when no intersections exist or can be found. A second click is required, as illustrated in Figure 11c. This processes are applied repeatedly to generate the desired roof boundary.

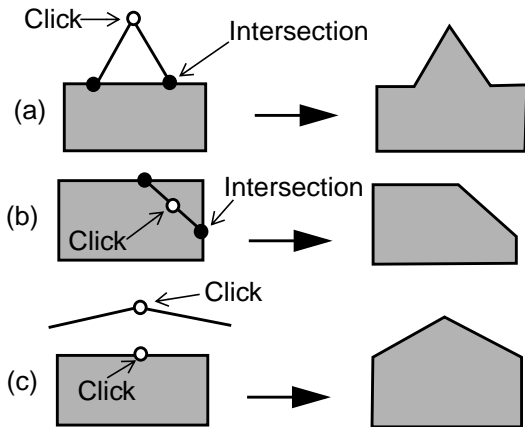


Figure 11. Adding/Subtracting a triangular block

Figure 12 shows an example of a building having a non-rectangular roof structure. The seed building shown in Figure 12a is generated by 3 clicks for an initial rectangular seed, followed by 2 clicks to remove a rectangular block on the top left. One additional click, near the indicated edge, is needed to remove the triangular block.

The height of the final model is taken to be that of the initial seed as it is correctly computed automatically, and no adjustment is needed. For complex buildings this type of interactions represent a typical case: The initial seed is

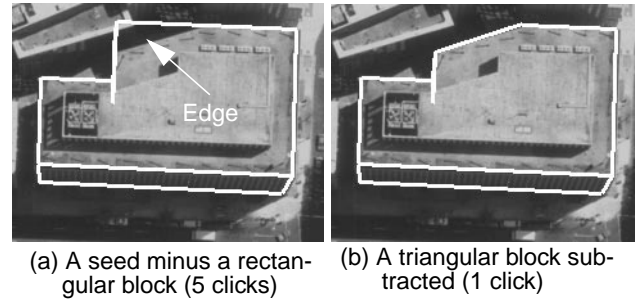


Figure 12. Non-rectangular building result

generated by three clicks, followed by additions and removals. When the seed is constructed by the automatic system, that saves 3 clicks but in some cases, however, the automatic seeds may require an editing step, such as adjustment of a corner or an edge.

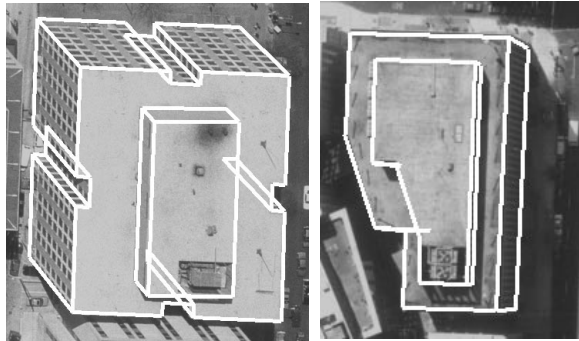
4.3. Multi-layer buildings

To convert a 2D roof into a 3D building model, the height for the roof is needed. This height is difficult to compute from the matched linear features underlying the roof hypotheses when more than one match is present. Instead, we conduct a fine search for a range of heights as follows: The 3D model is projected, at each height, to a second view and the supporting evidence for its roof is evaluated. The height value resulting in the best score is selected to be the height of the block. The local ground elevation is assumed to be known from an underlying DEM (digital elevation model), thus providing the actual height of the object above the ground.

This method has shown to be adequate, useful, and robust for the simpler, rectangular, single-layer structures and when it is preferred that all the building components and/or layers be extruded all the way to the ground level. In many cases, however, buildings can have multiple layers at different heights and extrusions of the higher layer to the ground are not correct or desired. Thus, a separate height calculation is required for each layer. The calculation of height for the lower layer proceeds in the manner described above. The elevation of the higher level proceeds similarly, with respect to the ground, but requires knowledge of the lower layer. This knowledge comes from two inclusion tests that determine whether the model under construction (higher layer) is "inside", in 2D, or "on top", in 3D, a previously constructed model. The 2D test is carried out for each vertex in the higher level to collect a list of roofs it is in, as more than one or more lower layers are possible. These are tested in 3D by projecting the roofs of the 2D selected models onto the ground and checking the roof of the higher layer under construction for full inclusion.

If no candidate is found, the process considers this layer to lay on the ground level. Also, we expect that higher layers may be smaller than the supporting ones and thus apply a finer height search. The actual height of the higher level is determined from the difference between the calculated height and the height of the supporting block.

Figure 13 shows some examples of the buildings shown earlier, that have a structure, or second layer, on top. Each level is processed independently, and in the same manner. No additional steps or clicks are needed to specify multiple layers; these are sought for and handled automatically.



(a) Two-layer building with structure on top (b) Non-rectangular building with two layers

Figure 13. Multi-layer buildings

4.4. Selection and verification

In the process of constructing models several possible configurations may exist at each step resulting in multiple hypotheses. We use two criteria to evaluate multiple hypotheses. The first one is based on determining self intersections among the model roof edges. Hypotheses that exhibit self intersections are topologically incorrect and are discarded. The second criterion evaluates the supporting evidence provided by the underlying features.

Handling self intersections.

When modeling multi-component buildings the topological ordering of the given and computed vertices, and on the underlying lines used by the system result in several possible configurations. These possibilities, in turn, lead to more than one possible hypotheses for a building block. Consider first the case of addition and removal of a rectangular block *A*, illustrated in Figure 14. The user does not have to specify addition or removal but merely that a modification is to be made. The system determines automatically which operation is taking place. In Figures 14a and 14b the user clicks near a junction (1) and on an interior point (2), a removal configuration. The order of clicking is not fixed. The ambiguity is in selecting L_I between the two alternatives to determine points 3 and 4 leading to two possible orderings of the four vertices. One of these result in

self-intersections, and the other (shown in Figures 14a and 14b) are plausible (depending on L_I).

In Figures 14c and 14d, the user clicks near a junction (1) and on an outside point (2), an addition configuration. The two choices for L_I result in two alternatives for the correct vertex ordering. One of these results in self intersections and the other (c or d) is correct depending on the choice of L_I .

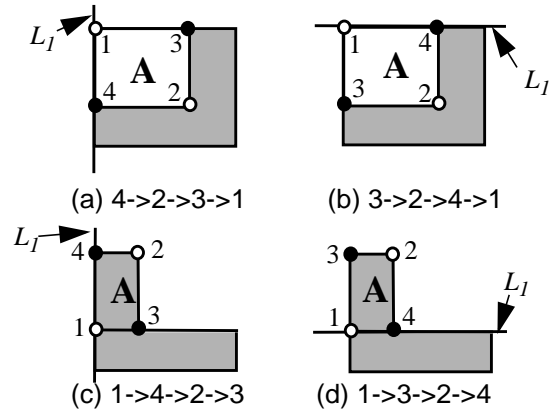


Figure 14. Vertex order ambiguity: (1,2) are user points, (3,4) are generated points

Handling multiple hypotheses.

For non-rectangular buildings, (Section 4.2), even when the user clicks near a good corner point, multiple edges near the clicked point may give rise to multiple hypotheses. Among these there may be more than one configuration that contains no self intersections. In these cases the system chooses the one that has the highest support from the underlying image lines. The measure of support is given in terms of the positive evidence (support) and negative evidence (conflict) provided by the underlying features. Positive edges are located near and parallel to the boundary of a roof hypothesis. Negative edges are also located near the boundary of a roof hypothesis but intersect the roof boundary (see Figure 15). The hypothesis that has the highest difference in the sum of the lengths of the positive and negative edges is chosen.

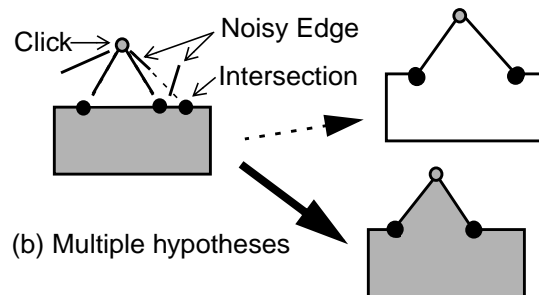


Figure 15. Choosing the stronger hypothesis

5. More results

We have tested our n method extensively. Figure 16 shows intermediate results while modeling a composite and irregularly shaped building. In this case the building is partially occluded by an adjacent building and its own top structure occludes part of the main layer (*i.e.* lower level building). The initial seed (3 clicks) plus a rectangular protrusion (2 clicks) are shown in Figure 16a. The addition of a triangular block (1 click) in Figure 16b is not sufficient to model the pointed protrusion due to its irregular angles. An additional triangular block is added (1 click) to conform to the shape shown in Figure 16c. Figure 16d displays the complete model with its top layer (3 clicks) which is followed by one adjust operation(2 clicks) for the occluded part of the main layer to be aligned with its top layer to make it reasonable structure.

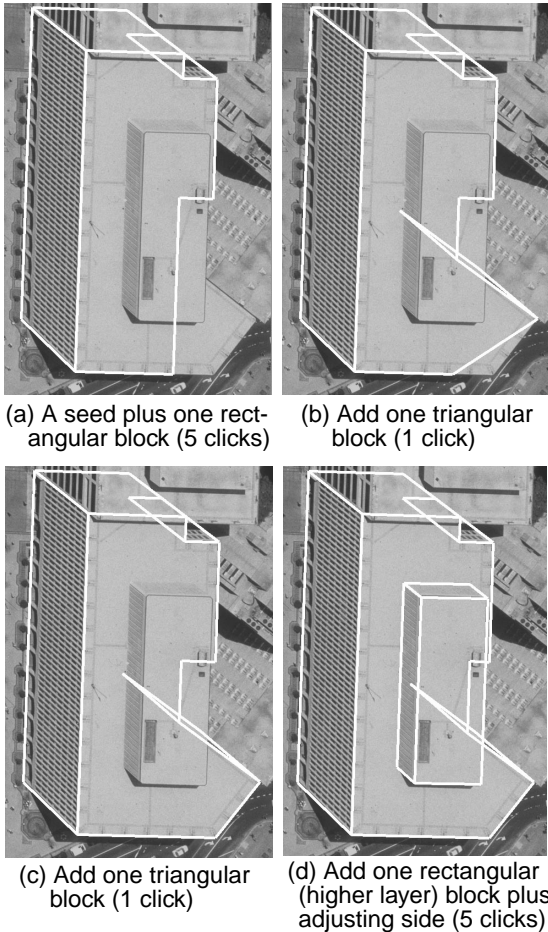


Figure 16. Generating irregular shape building with occluding areas; Total 12 Clicks

Another example is shown in Figure 17. A cluster of three buildings having irregular shapes occlude each other. The required user interaction is summarized in Table 2; it illustrates the number of clicks required for each

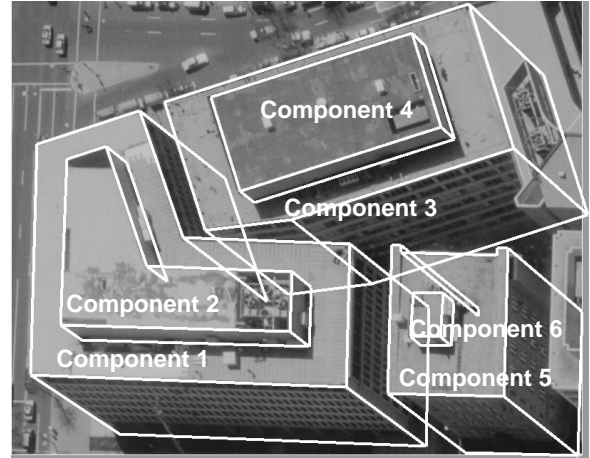


Figure 17. Building Cluster; 28 Clicks required to generate these models in 34 seconds

component, the amount of time for modeling each building and the total elapsed time.

TABLE 2. Statistics for Figure 17.

| Component | Clicks | Task | Time |
|--------------|-----------|----------------|-------------|
| 1 | 3 | Seed model | |
| | 1 | Subtract block | 6.0 |
| | 1 | Add block | |
| 2 | 3 | Add block | |
| | 1 | Subtract block | 6.0 |
| | 1 | Add block | |
| 3 | 3 | Seed model | 6.0 |
| | 2 | Subtract block | |
| 4 | 3 | Add block | 3.6 |
| 5 | 3 | Seed model | 6.0 |
| | 2 | Subtract block | |
| 6 | 3 | Add block | 6.0 |
| | 2 | Adjust height | |
| Total | 28 | | 33.6 |

The times reported in Table 2 do not include the setup time, the automatic processing time, nor the menu selection activity. The time needed for adjusting(or editing) buildings is however included. Also, the reported times represent averages over several trials as typically there are several ways to accomplish the same modeling tasks.

The buildings in this example were modeled by the minimum number of clicks after the seed building has been modeled by 3 clicks. Only 'Component 6' needed a height adjustment by 2 additional clicks.

An image of a University campus is shown in Figure 18a with 74 model buildings (wireframe) overlaid on the image. Figure 18b shows the same 3D models with their surfaces rendered and viewed from an arbitrary view-

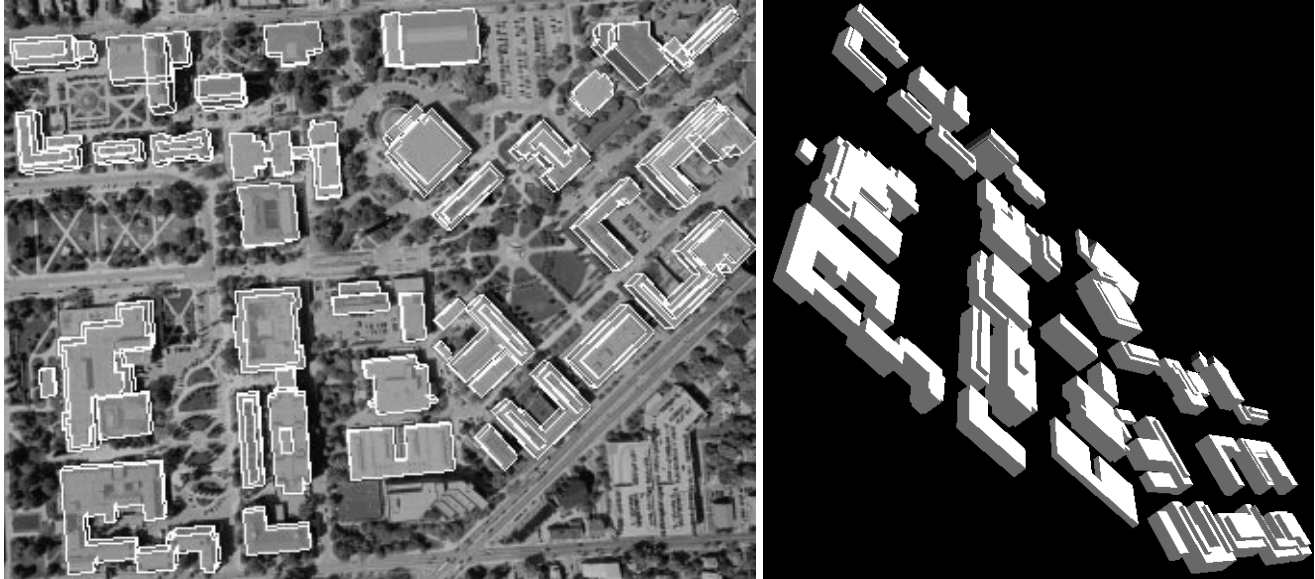


Figure 18. A site model of 74 buildings generated in 10 minutes (left) and an arbitrary 3D view of the site.

point. The seeds required 174 clicks, addition and subtraction of rectangular and triangular blocks required 168 and 6 clicks respectively. Corner removals (20 clicks), side adjustments (45 clicks), height adjustments (30 clicks) and angle adjustments (18 clicks) brings the total to 504 clicks required. At an average of 1.2 seconds of processing time per click, the time needed is 604.8 seconds, or about 10 minutes, of user wall time.

6. Conclusion

We have implemented, and demonstrated that critical user assists enable an automatic system to deliver high quality models. The added complexity of the models may still be challenging for completely automatic systems but can be achieved in a way that is considerably more efficient than traditional manual methods, at least by an order of magnitude. With the further addition of properly designed user interfaces and complete editing facilities, user-assisted systems can start to make it possible to construct models of cultural and other features in a less laborious and more productive ways. Future work will deal with an increased variety of roof surfaces and with curved structures.

7. References

- [1] A. Gruen and R. Nevatia (Editors), Special Issue on Automatic Building Extraction from Aerial Images, *Computer Vision and Image Understanding*, November 1998.
- [2] A. Gruen, E. P. Baltsavias and O. Henriksson, *Automatic Extraction of Man-Made Objects from Aerial and Space Images (II)*, Birkhauser Verlag, 1997.
- [3] J. McGlone and J. Shufelt, "Projective and Object Space Geometry for Monocular Building Extraction," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 54-61, 1994.
- [4] S. Noronha and R. Nevatia, "Detection and Description of Buildings from Multiple Aerial Images," *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, San Juan, PR, June 1997, pp. 588-594.
- [5] J. Li, R. Nevatia and S. Noronha, "User assisted Modeling of Buildings from Aerial Images," *Proceedings of the IEEE Computer Vision and Pattern Recognition Conference*, Fort Collins, CO, June 1999, pp. 274-279.
- [6] S. Heuel and R. Nevatia, "Including Interaction in an Automated Modeling System," *Proceedings of The DARPA Image Understanding Workshop*, Palm Springs, CA, February 1996, pp. 429-434.
- [7] T. Strat, L. Quam, J. Mundy, R. Welty, W. Bremner, M. Horwedel, D. Hackett, and A. Hoogs, "The RADIUS Common Development Environment," *Proceedings of the DARPA Image Understanding Workshop*, San Diego, CA, 1992, pp. 215-226.
- [8] Y. Hsieh, "SiteCity: A Semi-Automated Site Modeling System," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, San Francisco, CA, June 18-20 1996, pp. 499-506.
- [9] A. Gruen and H. Dan, "A Topology Builder for Automated Building Model Generation," in *Automatic Extraction of Man Made Objects from Aerial and Space Images (II)*, pp. 149-160, 1997.
- [10] E. Gülich, H. Müller, and T. Läbe, "Integration of Automatic Processes Into Semi-Automatic Building Extraction," *Proceedings of ISPRS Conference Automatic Extraction Of GIS Objects From Digital Imagery*, September 1999.